

Đèn trang trí

Một hệ thống gồm n đèn để tạo hiệu ứng ánh sáng. Các đèn được đánh chỉ số từ 1 đến n , mỗi đèn có 3 trạng thái, trạng thái sáng màu xanh hoặc sáng màu đỏ hoặc tắt. Ban đầu tất cả các đèn đều ở trạng thái tắt. Theo kịch bản sẽ có t lần thay đổi trạng thái của các đèn, lần thay đổi thứ k ($k = 1, 2, \dots, t$) sẽ thay đổi trạng thái của tất cả các đèn có chỉ số từ a_k đến b_k ($1 \leq a_k \leq b_k \leq n$). Với một đèn khi được thay đổi trạng thái sẽ thay đổi theo nguyên tắc như sau: nếu đèn đang ở trạng thái tắt sẽ chuyển sang trạng thái sáng màu xanh, còn nếu đang ở trạng thái sáng màu xanh thì chuyển sang trạng thái sáng màu đỏ, nếu ở trạng thái sáng màu đỏ thì chuyển về trạng thái tắt.

Ví dụ, nếu hệ thống gồm có 5 đèn và ban đầu đều ở trạng thái tắt, kịch bản gồm 3 thay đổi trạng thái các đèn, lần 1 thay đổi trạng thái các đèn có chỉ số từ 2 đến 4, lần 2 và lần 3 đều thay đổi trạng thái các đèn có chỉ số từ 3 đến 5. Khi đó, sau 3 lần thay đổi trạng thái của 5 đèn lần lượt là: tắt, sáng màu xanh, tắt, tắt, sáng màu đỏ.

Yêu cầu: Cho biết kịch bản gồm t lần thay đổi trạng thái của các đèn, lần thay đổi thứ k ($k = 1, 2, \dots, t$) sẽ thay đổi trạng thái của tất cả các đèn có chỉ số từ a_k đến b_k . Hãy cho biết, khi kết thúc buổi lễ thì có bao nhiêu đèn ở trạng thái tắt.

Dữ liệu: Vào từ file văn bản DLIGHT.INP có khuôn dạng sau:

- Dòng đầu chứa hai số nguyên dương n, t ;
- Dòng thứ k trong t dòng tiếp theo chứa hai số nguyên dương a_k, b_k ($1 \leq a_k \leq b_k \leq n$).

Kết quả: Ghi ra file văn bản DLIGHT.OUT gồm một số nguyên là số lượng đèn tắt khi buổi lễ kết thúc.

Ví dụ:

DLIGHT . INP	DLIGHT . OUT
5 3 2 4 3 5 3 5	3

DLIGHT . INP	DLIGHT . OUT
1000 1 2 999	2

Chú ý:

- Có 25% số test có $n \leq 10^6; t = 1$;
- Có 25% số test khác có $n \leq 10^3; t \leq 10^5$;
- Có 40% số test khác có $n \leq 10^6; t \leq 10^5$;
- Có 10% số test còn lại có $n \leq 10^9; t \leq 10^5$.

Camera quan sát

Có n camera quan sát, đánh số tuần tự từ 1 đến n , được đặt lần lượt thẳng hàng trên đường cao tốc từ Tây sang Đông. Camera thứ i được đặt ở độ cao h_i và hướng về một trong hai hướng: Tây hoặc Đông.

Camera thứ i có thể quan sát được camera thứ j nếu camera thứ i nhìn về phía camera thứ j và không có bất kỳ camera nào ở giữa hai camera này có độ cao vượt quá độ cao của camera thứ j .

Yêu cầu: Cho biết độ cao và hướng nhìn của n camera. Hãy cho biết mỗi camera có thể quan sát được bao nhiêu camera khác.

Dữ liệu: Vào từ file văn bản CAMERA.INP có khuôn dạng sau:

- Dòng đầu chứa số nguyên dương n ;
- Dòng thứ hai gồm n số nguyên dương h_1, h_2, \dots, h_n là độ cao của mỗi camera ($1 \leq h_k \leq 10^9$ với $1 \leq k \leq n$);
- Dòng thứ ba chứa một xâu gồm n ký tự cho biết hướng nhìn của các camera với quy ước: ‘W’ là nhìn sang hướng Tây, ‘E’ là nhìn sang hướng Đông.

Kết quả: Ghi ra file văn bản CAMERA.OUT trên một dòng n số tự nhiên, số thứ k cho biết số lượng camera mà camera thứ k ($k = 1, 2, \dots, n$) có thể quan sát được.

Ví dụ:

CAMERA . INP	CAMERA . OUT
5 1 2 2 3 1 EEWWE	3 2 1 2 0

Chú ý:

- Có 40% số test có $n \leq 10^2$;
- Có 20% số test có $n \leq 10^3$;
- Có 40% số test còn lại có $n \leq 10^5$.

Trò chơi đẩy bi

Trò chơi đẩy bi là một trò chơi trên lưới ô vuông vô hạn. Các dòng và cột của lưới được đánh số theo thứ tự bởi các số nguyên $\dots -3 -2 -1 0 1 2 3 \dots$. Các cột được đánh số theo thứ tự từ trái sang phải, còn các dòng theo thứ tự từ dưới lên trên. Ô nằm trên giao của dòng x và cột y được gọi là ô (x, y) . Khi bắt đầu chơi, người chơi được cho một số nguyên dương k và trên lưới sẽ xuất hiện một số viên bi, mỗi viên bi sẽ nằm gọn trong một ô và không có ô nào chứa nhiều hơn một viên bi. Người chơi sẽ phải chọn k ô phân biệt trên lưới làm ô hố, nếu ô được chọn làm ô hố có chứa bi thì viên bi đó sẽ biến mất. Sau đó, mỗi bước, người chơi có thể chọn một ô chứa bi và đẩy viên bi đó sang một trong bốn ô chung cạnh (hiện đang không có bi), nếu viên bi bị đẩy vào một trong k ô hố thì viên bi này sẽ biến mất. Nhiệm vụ của người chơi là đẩy hết tất cả các viên bi trên lưới vào hố với số bước ít nhất.

Yêu cầu: Cho biết vị trí các ô có chứa bi. Hãy chọn k ô tự do là ô hố và tìm cách đẩy tất cả các viên bi trên lưới vào hố với số bước ít nhất.

Dữ liệu: Vào từ file văn bản BGAME.INP bao gồm:

- Dòng đầu ghi hai số nguyên dương n, k ($2 \leq n \leq 12$);
- Dòng thứ i ($i = 1, 2, \dots, n$) trong n dòng tiếp theo, mỗi dòng chứa hai số nguyên x_i, y_i mô tả ô (x_i, y_i) là ô chứa bi.

Kết quả: Ghi ra file văn bản BGAME.OUT gồm một dòng chứa một số nguyên là số bước ít nhất cần thiết để đẩy tất cả các viên bi trên lưới vào hố.

BGAME . INP	BGAME . OUT
5 1 0 0 0 4 4 0 4 4 2 2	16

Chú ý:

- Có 20% số lượng test thỏa mãn điều kiện: $k = 1$ và các số x_i, y_i là số nguyên dương không vượt quá 100;
- Có 20% số lượng test khác thỏa mãn điều kiện: $k = 2$ và các số x_i, y_i là số nguyên dương không vượt quá 100;
- Có 20% số lượng test thỏa mãn điều kiện: $k = 1$ và các số x_i, y_i là số nguyên có trị tuyệt đối không vượt quá 10^6 ;
- Có 20% số lượng test khác thỏa mãn điều kiện: $k = 2$ và các số x_i, y_i là số nguyên có trị tuyệt đối không vượt quá 10^6 ;
- Có 20% số lượng test khác thỏa mãn điều kiện: $k \leq n$ các số x_i, y_i là số nguyên có trị tuyệt đối không vượt quá 10^6 .

ROBOT

Công ty X-TRANS đang sản xuất robot vận chuyển hàng hóa tự động trên mặt đất. Để làm việc đó, X-TRANS tiến hành huấn luyện robot trên một địa hình phẳng được chia thành một lưới các ô vuông gồm m dòng (đánh số từ 1 đến m) và n cột (đánh số từ 1 đến n).

Robot của X-TRANS có kích thước bằng đúng một ô vuông và có thể thực hiện các lệnh di chuyển đến các ô liền cạnh với ô đang đứng. Giả sử robot đang đứng ở ô (X, Y) , nó có thể thực hiện một trong bốn lệnh di chuyển sau:

- U: robot di chuyển đến ô $(X-1, Y)$
- D: robot di chuyển đến ô $(X+1, Y)$
- L: robot di chuyển đến ô $(X, Y-1)$
- R: robot di chuyển đến ô $(X, Y+1)$

	1	2	3	4	5
1					
2					
3					
4					
5					

Để mô hình huấn luyện gần với địa hình thực tế, X-TRANS đặt các vật cản tại p ô vuông phân biệt trên lưới địa hình để không cho robot đi vào. Robot chỉ có thể thực hiện một lệnh di chuyển nếu như ô mà nó di chuyển đến nằm bên trong lưới địa hình và không chứa vật cản. Nếu robot không thể thực hiện một lệnh, nó sẽ bỏ qua lệnh đó và tiến hành thực hiện lệnh tiếp theo.

X-TRANS tiến hành thử nghiệm robot với một tập gồm k lệnh để kiểm tra xem robot có thể di chuyển từ ô $(1,1)$ và kết thúc tại ô (m, n) sau khi thực hiện lần lượt các lệnh này được hay không. Nếu robot không kết thúc tại ô (m, n) , X-TRANS cần tìm cách xóa đi một số ít nhất các lệnh trong tập k lệnh để robot sẽ kết thúc tại ô (m, n) sau khi thực hiện tập các lệnh còn lại.

Giả sử robot đang đứng ở ô $(1,1)$ trên lưới địa hình được mô tả như hình trên (các ô màu xám chứa các vật cản), nếu robot thực hiện tập lệnh RDDDDRRRRRUD, nó sẽ kết thúc tại ô $(2,5)$. Để robot kết thúc tại ô $(5,5)$, X-TRANS cần xóa 3 lệnh để robot thực hiện tập lệnh còn lại là DDDRRRRRD.

Dữ liệu: Đưa vào từ file ROBOT.INP có cấu trúc như sau:

- Dòng đầu tiên là chứa 4 số nguyên: m, n, p, k ($2 \leq m, n \leq 40; k \leq 200$);
- Dòng thứ hai chứa một xâu gồm k kí tự thể hiện k lệnh điều khiển robot;
- p dòng tiếp theo, mỗi dòng chứa 2 số nguyên dương x và y cho biết có đặt một vật cản tại ô (x, y) . Không đặt vật cản tại hai ô $(1,1)$ và (m, n) .

Kết quả: Đưa ra file ROBOT.OUT chứa một số nguyên là số lượng ít nhất các lệnh cần xóa để robot sẽ kết thúc tại ô (m, n) sau khi thực hiện tập các lệnh còn lại. Nếu không tồn tại cách xóa để di chuyển robot đến ô (m, n) thì ghi ra -1.

ROBOT.INP	ROBOT.OUT
5 5 2 12 RDDDDRRRRRUD 2 2 5 3	3